LEVEL II

2

AFRPL-TR-81-69

Thiokol Report
U-81-17A

AD A108634

FINAL TECHNICAL REPORT

ADVANCED AIR LAUNCHED MISSILE MOTOR DESIGN METHODS

G. P. Roys

Thiokol Corporation
Huntsville Division
Huntsville, Alabama 35807

September 1981

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The AFRPL Technical Services Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

Prepared for

AIR FORCE ROCKET PROPULSION LABORATORY
DIRECTOR OF SCIENCE AND TECHNOLOGY
AIR FORCE SYSTEMS COMMAND
EDWARDS AFB, CALIFORNIA 93523

NOTICES

When U. S. Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, or in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that way in any way be related thereto.

FOREWORD

This report was submitted by Thiokol Corporation/Huntsville Division, Huntsville AL 35807, under Contract F04611-80-C-0016, Job Order No. 314809VG with the Air Force Rocket Propulsion Laboratory, Edwards AFB, CA 93523. This Technical Report is approved for release and distribution in accordance with the distribution statement on the cover and on the DD Form 1473.

John H. Costello Jr.
JOHN H. COSTELLO, JR, 1Lt, USAF
Project Manager

Lee G. Meyer
LEE G. MEYER, Chief
Air-Launched Missile
Propulsion Branch

FOR THE COMMANDER

Charles R. Cooke
CHARLES R. COOKE
Director, Solid Rocket Division

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFRPL-TR-81-69	2. GOVT ACCESSION NO. AD-A108634	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Advanced Air Launched Missile Motor Design Methods, Final Report		5. TYPE OF REPORT & PERIOD COVERED Final Report 21 Jan 80 thru 4 Sep 81
		6. PERFORMING ORG. REPORT NUMBER U-81-17A
7. AUTHOR(s) G. P. Roys		8. CONTRACT OR GRANT NUMBER(s) AFRPL F04611-80-C-0016
9. PERFORMING ORGANIZATION NAME AND ADDRESS Thiokol Corporation Huntsville Division Huntsville, AL 35807		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS JON: 314809VG
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Rocket Propulsion Laboratory/MKAS Edwards Air Force Base, CA 93523		12. REPORT DATE September 1981
		13. NUMBER OF PAGES 42
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) N/A		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release. Distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Approved for public release. Distribution unlimited.

18. SUPPLEMENTARY NOTES

Principal Investigator for Thiokol was G. P. Roys. W. I. Dale, Jr. was Program Manager. Significant technical contribution was made by W. C. Aycock, D. D. Maser, P. R. Renfroe, and R. O. Hessler.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Solid propellant rocket motor, mathematical modeling, numerical non-linear optimization.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report is the final report of a three-phase project whose objective was to formulate a computer code to perform detailed preliminary designs of solid propellant rocket motors. All major components and performance of a motor are mathematically modeled using source dimensions and characteristics. A direct pattern search non-linear optimization scheme based on the Hooke and Jeeves algorithm is employed to establish motor characteristics that optimize

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified 401944

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. any one of several performance parameters. Constraints imposed during the optimization process are basic performance requirements, design constraints and operating limits. Decision variables during optimization are propellant formulation, propellant burn rate, propellant grain dimensions, nozzle dimensions, and pressure vessel dimensions. Provisions are made for easily inserted user-defined models of several characteristics.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

As part of the Low Cost Missile Motor Demonstration Program (Reference 1)*, tradeoffs between performance, reliability, and cost were performed. Under that program, a ballistic simulation code was combined with a non-linear direct-pattern search routine and a subroutine that incorporated various cost models, design constraints and performance requirements. The new design integration code (Thiokol designation E469) described a motor whose cost or, alternately, weight was minimized. Propellant geometry, propellant formulation, and motor geometry were adjusted until motor performance requirements, design constraints, propellant constraints and combustion stability criteria were satisfied.

During preparation and subsequent use of the computer code, two areas of potential improvement were identified that would make the design technique more universal by including (1) more general motor configurations and propellant formulation capabilities, (2) more detailed and precise design constraints in the areas of propellant structural loads, propellant thermochemical characteristics, nozzle geometric relations, case/closure interfaces, and combustion stability.

PROJECT OBJECTIVE

The overall objective of this project was to update and improve the existing E469 computer code into an advanced code for optimizing tactical and strategic air-launched missile motor designs.

PROJECT DESCRIPTION

Work was accomplished in three phases. Phase I (Code Preparation) was divided into three tasks and included all activities required to develop the computer code. Task 1 defined the level of program detail; Task 2 was model formulation; and Task 3 consisted of computer code preparation. Phase II (Documentation) saw computer code documentation in the form of user manuals. Phase III (Demonstration and Evaluation) demonstrated computer code operation (using several AFRPL-selected design problems) and included an Industry Briefing.

*References are listed at the end of the report.

PHASE I: COMPUTER CODE PREPARATION

Task 1: Determination of Level of Detail

The purpose of Task 1 was to provide Thiokol and AFRPL an overall plan for the computer code prior to the expenditure of major funds and calendar time on the details of modeling and coding. This task was used to establish basic approaches. Qualitative trade-offs were made between modeling accuracy, program execution time, and central memory requirements. In general, it was determined what the code would provide in the way of features and how those features would be modeled. Choices were made between analytical and empirical models for each computation module.

Results of this effort were documented in a formal report (Reference 2) which served as the detailed plan for the code development. A meeting between Thiokol and AFRPL personnel served to clarify certain points, and several additions to the plan were incorporated. Formal (PCO) approval was received to follow the Reference 2 plan.

During Task 1, the existing computer code (E469) was formally submitted to AFRPL, along with a User's Manual (Reference 3).

Task 2: Model Formulation

This task involved the development of the mathematical models that describe the motor components and performance characteristics. The plan set forth in Reference 2 was followed almost exactly. All changes to the modeling approach were keyed to the Reference 2 plan.

Task 3: Computer Code Preparation

Models developed under Task 2 were reduced to coding as subroutines (or as groups of subroutines) in this task. Task 2 and Task 3 were conducted concurrently except for a short time at the beginning of Task 2 and at the end of Task 3. The procedure followed was to formulate the details of a particular model, reduce it to computer coding, and then perform subroutine and module testing. At a certain point during Task 3, modules were combined to provide the skeleton of the final code to allow testing of the overall arrangement. Then additional modules were added as they were available following module-testing.

PHASE II: DOCUMENTATION

Task 1: User's Manual

A three-volume User's Manual (Reference 4) was prepared during this task. Volume I (Technical Description) gives the basis for the code

computations analytical developments, logic flow charts used in verification checks and error messages. Volume II (User's Guide) contains the input and output dictionaries and their accompanying illustrations, along with other input instructions needed to execute the code. Volume III (Code Description) contains the subroutine descriptions and flow charts and cross-indices of common statements, subroutines and call statements.

The User's Manual is arranged by major sections and the numbering system of pages, figures and tables follows the same section system. Thus, future modifications to the code can be (should be) easily documented by revising the appropriate section of the Manual.

Task 2: Program Compatibility

The objective of this task was to insure that code programming was compatible with other machines to minimize difficulties in transferring the code to other machines.

All programming was in FORTRAN IV language according to ANSI standards. The code is operational on IBM 4341 and CDC6600 computers, with minimum conversion required for operation on the two machines; in fact, there are only two differences in the code for the two-computers. Double-precision statements are required on the 16-bit IBM computer and they are installed in the basic code; however, double-precision provisions are not needed on the 32-bit CDC computer, and so the statements are left in place but are disengaged with "comment" notations. Arc sin and arc cos functions are named ARSIN and ARCOS, respectively, in the IBM version and ASIN and ACOS in the CDC version; however, the latter can be used directly on an IBM computer employing the H-extended compiler.

Task 3: Computer Program Model Review

The objective of Task 3 was to provide periodic review of the computer code development so that a product would be delivered to AFRPL that met all project objectives.

A meeting was held at AFRPL at the end of Phase I, Task 1 to review the report (Reference 2) that would guide code development. During conduct of Phase I, Task 2 and Task 3, Thiokol and AFRPL personnel held face-to-face review meetings on a regularly scheduled basis (every three months), during which current status was discussed in detail and future activities were described. There were usually several minor points that were clarified and minor adjustments made in the technical approach. These reviews, along with frequent telephone contacts and the monthly progress reports, were instrumental in preventing any major project redirection by insuring that Thiokol was providing what AFRPL wanted.

The final code and User's Manual were submitted to AFRPL at a final review at the end of Phase I, Task 3.

PHASE III: DEMONSTRATION AND EVALUATION

In Phase III, the operational state of the computer code was demonstrated by solving five AFRPL-supplied sample problems (Task 1). In addition, an Industry Briefing was held during this phase to teach potential industry users how to use the computer code (Task 2). This briefing was followed by a two-month evaluation period in which users of the code were given additional aid in code use plus any assistance necessary to resolve problems brought out by the users (Task 3).

COMPUTER CODE DESCRIPTION

SUMMARY

The Solid Propulsion Optimization Code (SPOC) performs detailed preliminary designs of a large variety of solid propellant rocket motors. Dimensions of the propellant grain, nozzle, and pressure vessel are adjusted by the code, along with propellant formulation and burn rate, to produce a motor design that meets performance requirements and satisfies design constraints and operating limits and that has been optimized with respect to a performance parameter selected by the user from a menu.

SPOC was prepared for use by a motor designer. The user/designer controls the direction taken by the search through the inputs. Information used in the code must be provided by the designer, but no more is required than what must already be accumulated in order to prepare a detailed preliminary design---which is what this code will produce. It is not intended that this code replace final detailed stress, thermal, and combustion stability analyses; it will monitor certain stress, thermal and stability parameters in the search for an optimized design so that the final arrangement is more likely to pass detailed analyses. SPOC will do no more, nor will it do any less, than a good designer will do; but the code will do it much faster, thus enabling the designer to examine more approaches and more combinations than previously possible.

The user supplies a starting condition---an initial design---and all associated information needed to evaluate that design. These data are read in through a series of input namelists. The initial design is evaluated in the executive subroutine COMP and the results are printed. Then the print control is turned off[1] and optimizer routine (PATSH) is called, which in turn makes multiple calls to COMP to evaluate the design with PATSH-generated changes in values of user-specified parameters. Once an optimized design[2] is reached, the print control is turned on and a final pass through COMP gives a complete description of the final design.

(1) There is a control that specifies a complete print-out of all design analyses for each pass through COMP, which is useful for determining why a search is behaving as it does, but which produces a great deal of printout.

(2) An "optimized design" is reached when no further improvement in the objective function is realized, as defined by: (1) the number of base points defined in the search equals an input limit; (2) the pattern search step size becomes less than an input minimum. Either of these two conditions will trigger a return to COMP for the final evaluation.

MOTOR AND PROBLEM DEFINITION

SPOC includes models for five propellant grain configurations, three forward closure and two aft closure arrangements, and six nozzle configurations. Any combination of grain, closure, and nozzle may be selected except that a Type 4 grain (conocyl) may be used only with the Type 1 forward closure (ellipsoidal).

The analyses performed by SPOC are

Thermochemistry	Pressure vessel structural
Ballistic	Nozzle thermal and structural
Propellant structural	Trajectory
Weight	Combustion stability
Cost	Impulse efficiency

Flexibility has been provided for the user/designer so that the code may be tailored to enable varied problems to be solved. These choices are described in the following paragraphs.

Propellant Grain: choose one from those illustrated in Figure 1.

Type 1: Star
Type 2: Double-web wagon wheel
Type 3: Finocyl (slots in forward end)
Type 4: Conocyl
Type 5: Cylindrically perforated (CP)

Nozzle: choose one from those illustrated in Figure 2.

Type 1: Thin shell, composite structure as the insulating ablative and support structure.

Type 2: Thin shell support structure with throat insert and ablative insulator.

Type 3: One-piece ablative; supersonic blast tube; constant diameter support structure.

Type 4: One-piece ablative; supersonic blast tube; reduced diameter aft section.

Type 5: Subsonic blast tube; without expansion cone.

Type 6: Subsonic blast tube; with expansion cone.

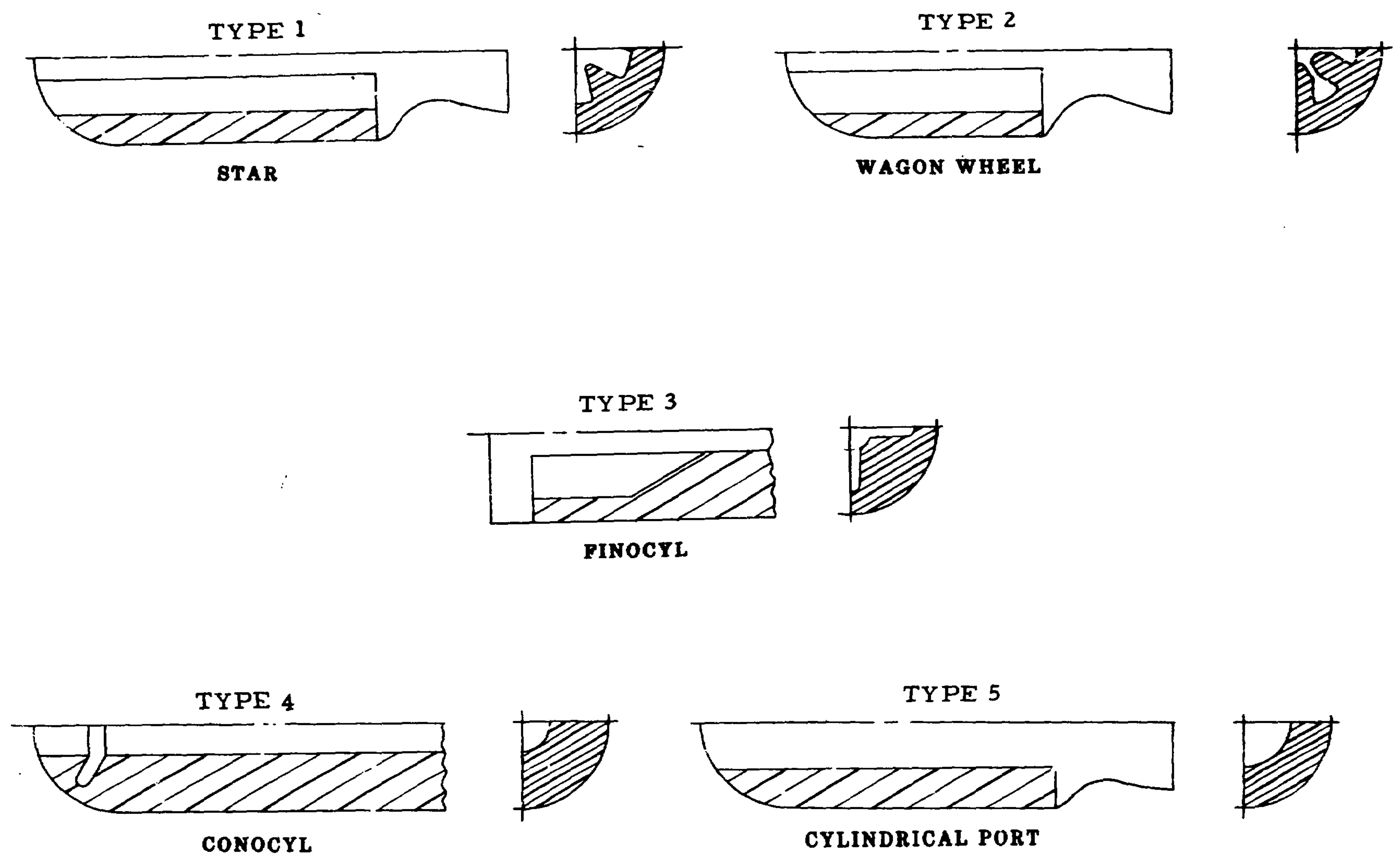
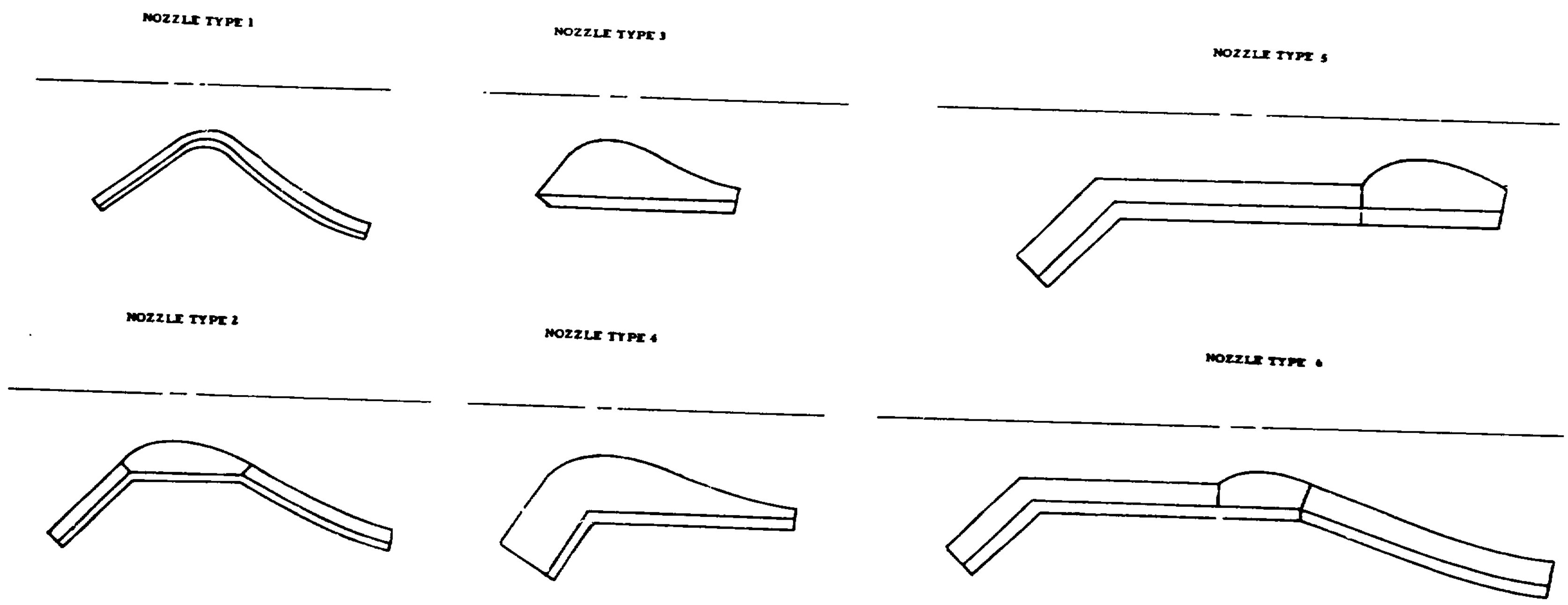


Figure 1. Propellant Grain Geometries Available in SPOC



All Exit Sections May be Conical or Contoured

Figure 2. Nozzle Configurations Available in SPOC

Forward Closure: choose one from those illustrated in Figure 3.

Type 1: Ellipsoidal

Type 2: Flat plate with closure secured with retaining ring

Type 3: Flat plate with closure integral with case

Aft Closure: choose one from those illustrated in Figure 4.

Type 1: Ellipsoidal

Type 2: None (aft closure formed by nozzle entrance section)

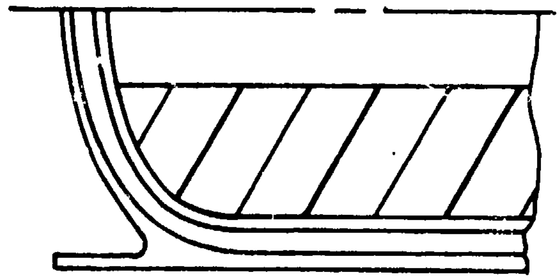
Other choices that must be made to define the problem are:

1. A propellant formulation may be input and adjusted as part of the optimization (FORMAD=T), in which case the thermochemistry routines are entered every time the design is evaluated (except for some internal by-passes to reduce execution time). Another option is to input a formulation but not adjust it (FORMIN=T), in which case the thermochemistry routines are entered only for the first evaluation in order to obtain basic propellant characteristics for the ballistic simulation. The third option is for the user to input the appropriate ballistic parameters rather than having the thermochemistry routines calculate them from a formulation (PROPIN=T). The proper combination of these three inputs is shown below (all default to F).

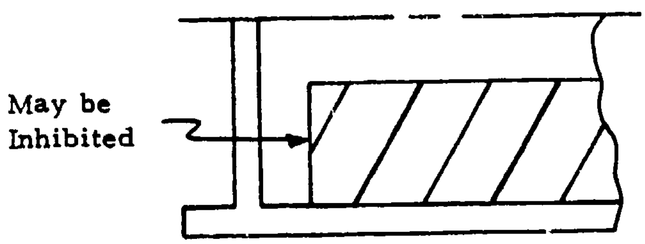
	MODE	FORMAD	FORMIN	PROPIN
(1)	Formulation input and adjusted during optimization	T	F	F
(2)	Formulation input, but not adjusted	F	T	F
(3)	User supplies required propellant characteristics	F	F	T

2. Impulse efficiency may be input by the user, calculated internally with the AFRPL SPP "empirical model" (Reference 5), or calculated with a user-supplied model which must be installed in subroutine USEREF. EFMDL=T is the flag to show a user model has been supplied. SPPETA=T is the flag to specify the SPP model.

3. Propellant burn rate is calculated internally with the Vielle model, or with a user-supplied model which he must install in subroutine USERRB. RBMDL=T is the flag to show a user model has been supplied.

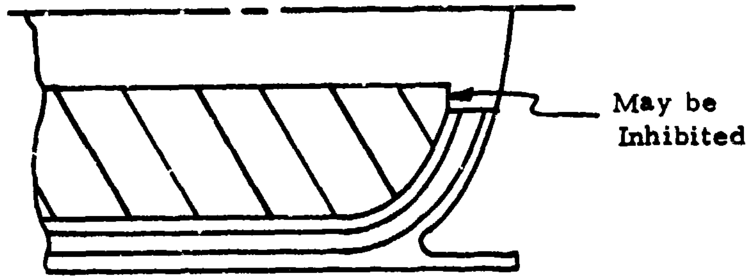


Ellipsoidal - Type 1

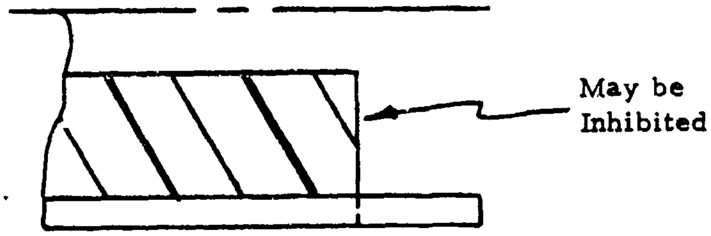


Flat Plate - Type 2 (Retaining Ring)
- Type 3 (Integral with Case)

Figure 3. Forward Closure Configurations Available in SPOC



Ellipsoidal - Type 1



None - Type 2 (Formed by Nozzle)

Figure 4. Aft Closure Configurations Available in SPOC

4. The propellant face on the forward end of a grain with a Type 2 or Type 3 forward closure and on the aft face of a grain with either a Type 1 or Type 2 aft closure may be inhibited through use of FWDINH=T or AFTINH=T, respectively.

5. Ballistic simulations will be performed at both the low temperature and high temperature conditions if different values are input for THI and TLO. Propellant structural analysis is performed at a different temperature (TPROP) than is the low temperature ballistic simulation. Pressure vessel structural analysis is performed at the high temperature condition. If THI is input equal to TLO, only one ballistic simulation is performed; propellant structural analysis is still performed at TPROP. Pressure vessel structural analysis is performed with the results of the single-temperature ballistic simulation (i.e., pressure not adjusted to some high temperature condition).

6. The optimization routine will adjust user-specified parameters in order to meet all performance requirements and satisfy all design constraints. In addition, the user may specify another parameter to be optimized by setting ICHOZE to one of the following.

0:	None (default value)
1:	Minimize cost
2:	Minimize total motor weight
3:	Maximize total impulse
4:	Maximize total impulse-to-total weight ratio
5:	Maximize burnout velocity

7. There are 36 parameters (not all on one problem) whose values can be adjusted by the optimization routine PATSH to achieve an optimum design (Table 1). Each of these must be specified by the user as "T" (maintain at input value) or "F" (do not maintain at input value, but adjust during pattern search). Default value is T (do not adjust).

8. A trajectory simulation (point mass, flat earth, ballistic trajectory) will be performed if specified by the user (FTRAJ=T). If ballistic simulations are performed at two temperatures (TLO and THI), then trajectory simulations are performed with each of the resultant thrust-time histories. In addition, the user must select a trajectory termination option.

9. With FCOST = T, motor cost will be calculated using either the Tri-Services cost model (Reference 6) or a user-supplied model. CSTMDL = F is the flag to specify the Tri-Services model; CSTMDL = T is the flag to show a user-supplied model has been provided.

10. Either a contoured or conical nozzle expansion section may be specified (CONTUR=T or CONTUR=F, respectively). If a conical

TABLE 1

ADJUSTABLE VARIABLES AVAILABLE IN SPOC

Variable Number	Variable Name	Definition
1	BIND	Weight fraction of binder
2	FUEL	Weight fraction of fuel
3	OXA(1)	Weight fraction of oxidizer A, Size (1)
4	OXA(2)	Weight fraction of oxidizer A, Size (2)
5	OXA(3)	Weight fraction of oxidizer A, Size (3)
6	OXB(1)	Weight fraction of oxidizer B, Size (1)
7	OXB(2)	Weight fraction of oxidizer B, Size (2)
8	OXB(3)	Weight fraction of oxidizer B, Size (3)
9	RCATL	Weight fraction of liquid rate catalyst
10	RCATS	Weight fraction of solid rate catalyst
11	STAB	Weight fraction of combustion stabilizer
12	RE	Nozzle exit radius (Note 1)
13	RT	Nozzle throat radius (Note 1)
14	R2A1	Propellant port radius at Plane 1 (Grain 3), and star point tip radius at Plane 1 (Grain 2)
15	R2A14	Propellant port radius at Plane 14 (Grains 3, 4 and 5)
16	R4A1	Propellant slot fillet radius at Plane 1 (Grain 3)
17	R5A1	Propellant slot depth radius at Plane 1 (Grain 3) and fillet radius between propellant tip and web (Grain 1 and 2)

(1) Input as appropriate diameter

Table 1

ADJUSTABLE VARIABLES AVAILABLE IN SPOC (contd.)

Variable Number	Variable Name	Definition
18	ALPHA1	Angle on side of propellant slot (Grain 3), and included half-angle of star tip (Grain 1)
19	LSLOT	Length of propellant slot (Grain 3)
20	LCP	Length of cylindrically perforated grain section (Grains 3, 4, 5)
21	LCONE	Length of aft coned grain section (Grains 3, 4, 5)
22	ALFAEX	Nozzle exit half angle
23	TCASE	Case wall thickness, cylindrical section
24	RB70	Propellant burn rate at 70°F, 1000 psia
25	XN	Pressure exponent in Vielle burn rate model
26	LCONEF	Length of forward coned grain section (Grain 5)
27	RMOTOR	Motor outside radius (Note 2)
28	R2A3	Propellant port radius at Plane 3 (Grain 4)
29	RTIP	Outboard radius of propellant slot (Grain 4)
30	ZED	Angle of slot with centerline (Grain 4)
31	LH	Length of forward propellant segment (Grain 4)
32	TAUW1	Propellant web thickness (Grains 1 and 2)
33	LSA1	Star tip height at Plane 1 (Grains 1 and 2)
34	LSA14	Star trip height at Plane 14 (Grains 1 and 2)
35	LFWD	Length of forward untapered propellant section (Grains 1 and 2)
36	LTAPER	Length of aft tapered propellant sections (Grains 1 and 2)
37	R2A5	Propellant port radius at Plane 5 (Grain 5)

exit section is selected, the initial half-angle of the expansion section (ALFA) must be input equal to the exit half-angle (ALFAEX).

11. Several analyses are by-passed completely unless the user specifies otherwise.

(a) Propellant structural analysis (PSTRUC=T)
(b) Combustion stability (FSTAB=T)
(c) Trajectory simulation (FTRAJ=T)
(d) SPP impulse efficiency (SPPETA=T)
(e) Thermochemistry (FORMAD=T or FORMIN=T)
(f) Cost (FCOST=T)

12. The user may provide models for certain parameters that are used in the analyses. A flag is set to show a user model has been loaded into a specified subroutine (T=model has been supplied).

Flag	Load in Subroutine	Parameter to Be Supplied
RBMDL	USERRB	Propellant burn rate, RATE (in/sec)
SEMDL	USERSE	Propellant nominal strain endurance, SENOM (in/in)
EOMMDL	USERRH	Propellant rheological property to be defined by user, EOM (units by user)
CSTMDL	USERCS	Motor cost, COST ($ or $/unit)
EFMDL	USEREF	Impulse efficiency, ETAISP (% x 0.01)
*	RSPNSE	Combustion response

*IRSPNS = 5 in namelist STABIN

13. If a combustion stability analysis (Reference 7) is desired, the user can select one of five combustion response models (one is user-supplied) and can specify at how many modes stability margin is to be calculated.

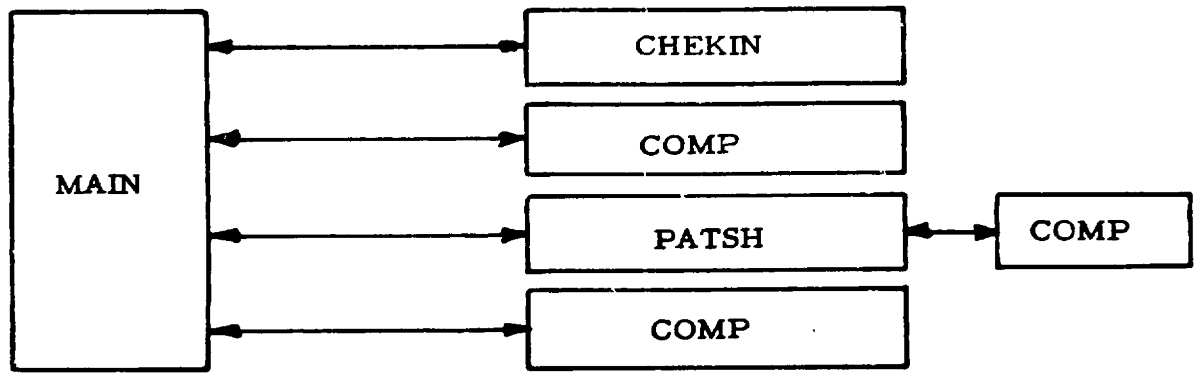
COMPUTER CODE ARRANGEMENT

The computer code has an overall organization shown in Figure 5. There are three major subprograms (MAIN, COMP, and PATSH), whose functions are listed in Figure 5. MAIN first reads and initializes various parameters and calls subroutine CHEKIN to verify the compatibility of the problem defined by the user and to print a narrative description. A call is then made to COMP for the first time in order to calculate performance of the motor with user-supplied initial values. Initial values of penalties are also calculated (in COMP) and all output is printed, after which the flag is turned off. MAIN then calls PATSH to adjust specified parameters in order to minimize the payoff parameter and penalties. Each time PATSH adjusts one or more of the specified parameters, COMP is called to calculate motor performance, payoff and associated penalties. PATSH builds a pattern and makes adjustments to minimize the OBJ function. When there is no further decrease in the payoff and penalties, the flag is turned on, COMP calculates the performance with the last set of adjusted parameters, and results are printed.

The executive subroutine COMP sets up the user- or PATSH-supplied inputs for the various analyses and simulations and passes the results of early analyses to later calculations when they are needed (Figure 6). For the first pass through COMP, where all analysis inputs are furnished by the user, the inputs are read in the specific subroutine to which the data apply. On all subsequent passes through COMP, the input data are either constant at the user-input value or are updated according to the PATSH adjustments. Write commands are also given within the individual subroutines.

The first call by COMP is to one of the grain dimension verification and setup subroutines (SETUP1 for Grain Type 1, SETUP2 for Grain Type 2, etc.): these subroutines verify the geometric validity of the incoming dimension set and calculate other dimensions needed by the ballistic simulation module. Subroutine NOZINP is called to perform the same function for the nozzle. If the problem involves a propellant formulation, subroutine TCHEM is called next to perform thermochemical analyses; results of the calculations are used in IMPEFF (impulse efficiency), SEC2SB (ballistic simulation), NOZL (nozzle thermal and structural analysis), and E488M2 (combustion stability). Subroutine IMPEFF is called next to furnish a value for impulse efficiency, if specified by the user.

Subroutines SEC1SB and SEC2SB make up the ballistic simulation module. The first time they are called, the input ballistic parameters have been set up (in COMP) for a grain conditioned to high temperature conditions. When the ballistic simulation is completed, subroutine HITEMP uses the results to calculate certain performance parameters and operating conditions associated with high temperature motor operation (e.g., design pressures, minimum burn time, etc). The predicted values are



MAIN: Reads control inputs; initializes some parameters; controls printout; calls search routine

COMP: Executive subroutine passes information between subroutines; calculates some penalties and overall objective function (OBJ); provides printout

PATSH: Adjusts specified parameters; evaluates changes in objective function (OBJ) to optimize

Figure 5. Overall Code Arrangement

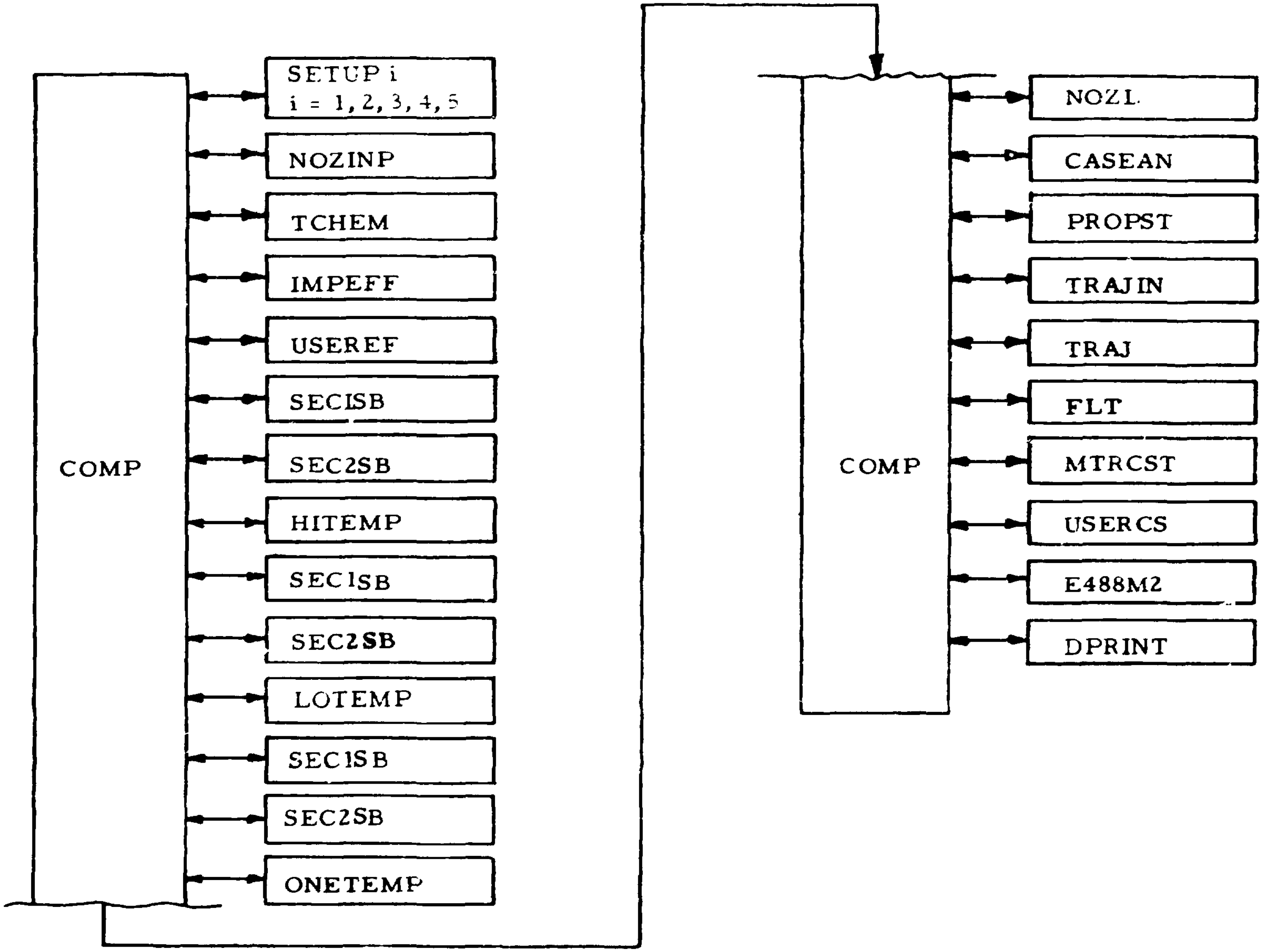


Figure 6. Subroutine COMP

compared with user input limits and appropriate penalties are calculated. Next COMP sets up ballistic parameters for a simulation with the grain conditioned to low temperature, and then SEC1SB and SEC2SB are called again. Results of the low temperature simulation are analyzed in subroutine LOTEMP for performance parameters and operating conditions associated with low temperature motor operation. If the user wants only to study a problem via ballistic simulation at a single temperature, the second simulation is skipped (by making THI = TLO) and results of the first are analyzed in subroutine ONETMP (that combines the calculations of HITEMP and LOTEMP).

Once the results of the ballistic simulations(s) are available, nozzle thermal and structural analyses are performed in subroutine NOZL, pressure vessel structural analyses are performed in subroutine CASEAN, and (if specified by the user) propellant structural analyses are performed in subroutine PROPST. The user may also command a trajectory simulation. Subroutine TRAJIN acts as a mini-executive subroutine to control the trajectory simulations for a one-or two-temperature problem. Motor cost is calculated in subroutine MTRCST, and combustion stability characteristics are determined in subroutine E488M2, if specified by the user.

OPTIMIZATION PROCESS

SPOC combines computer models for solid rocket motor performance prediction and design analyses with a numerical parameter optimization technique. As stated in Reference 8, this combination requires an understanding of both areas. The following discussion was taken from Reference 8 because approaches taken in the TACMOP and SPOC codes are very similar, even though the codes have different end objectives.

In order to eliminate misinterpretation, several terms used throughout the remainder of the discussion are defined below:

Performance requirement - A measure of acceptable system operation in accomplishing its intended purpose. For solid-propellant rocket motors, performance requirements typically include such items as range, velocity, or payload delivered to a specified end condition. In SPOC, performance requirements are expressed as total impulse, impulse-to-weight ratio, etc., as well as the ultimate end-item requirements listed above; however, the trajectory simulation in SPOC is not intended for complex maneuvering trajectories, and so SPOC should be used in conjunction with more sophisticated trajectory simulations.

Design parameter - A length, angle, or material property used in describing a particular design, such as propellant grain length, case diameter, nozzle half angle, or propellant burning rate.

Design constraint - A limit imposed directly or indirectly on the allowable values of a design parameter, such as maximum length, maximum nozzle divergence angle, maximum propellant web fraction, or minimum port-to-throat area ratio.

Operating limit - A maximum or minimum acceptable level for a condition produced by motor operation, such as maximum acceleration, minimum pressure, or maximum velocity.

Payoff - The quantity selected as the maximized or minimized variable during the optimization process, such as maximum range. In SPOC, corresponding payoffs are total impulse, motor weight, cost, etc.

Penalty function - A function corresponding to a particular performance requirement, design constraint, or operating limit, having zero value when the requirement, constraint, or limit is satisfied by the design being evaluated, and having a non-zero value proportional to the amount of violation of the particular requirement when it is not satisfied.

Objective function - A single-valued function for a particular design representing both the payoff value and any non-zero penalty function values associated with that design.

The design problem consists of finding a set of design parameter values that produce a system with maximum (or minimum) payoff, subject to meeting all performance requirements, design constraints, and operating limits (i.e., all penalties non zero).

Parameter Optimization Scheme

The optimization routine used in SPOC is the PATSH (Pattern Search) subroutine developed by D. E. Whitney at the Massachusetts Institute of Technology (Reference 9). This subroutine performs an unconstrained non-linear optimization with the direct pattern search algorithm of Hooke and Jeeves (Reference 10). This particular scheme has delivered good performance when compared with other methods (Reference 11 and 12). Direct search methods operate on the basis of always saving the most optimum point encountered as the new "base point", or point about which further searches are made.

The Hooke and Jeeves direct search is unconstrained in itself; however as applied here the problem is constrained through the manner in which the single-valued objective function (OBJ) is calculated. Limits on the magnitude of the decision variables, as well as analytical relationships between the decision variables, are imposed through the use of individual penalties.

PATSH operates by "moving" (adjusting) the decision variables

$$\underline{X}^{i+1} = \underline{X}^{i} + (0.05)(\mathrm{DEL})\,\underline{X}^{i}$$

where $\underline{X}^{i}$ = current decision variable set

$\underline{X}^{i+1}$ = new decision variable set

DEL = step size multiplier

These are two results of moves. A <u>successful</u> move produces a reduction in the objective function OBJ. A move is a <u>failure</u> when there is no reduction in OBJ. Moves can be accomplished in one of two ways. An <u>exploratory move</u> consists of changing the value of only one decision variable and evaluating OBJ. A <u>pattern move</u> occurs when values of all decision variables are changed simultaneously according to the information derived from exploratory moves. During a pattern move, each variable is changed by an amount proportional to the difference between its value at the current base point and its value at the immediately preceeding base point.

The logic flow of PATSH is presented in Figure 7. PATSH begins the search by calling the computational program (subroutine COMP) with the initial user-supplied parameter set to establish the initial base point; this produces an analysis identical to the first call to COMP by MAIN. In the call to the computational package, PATSH sends the current parameter set to the package and receives back the objective function value corresponding to that parameter set. After evaluation of the initial base point, PATSH begins a series of exploratory moves, varying the value of each parameter in the following systematic manner:

(1) Vary the parameter in the positive direction by five percent and evaluate the objective function. If the objective function decreases in value from the base point, keep the parameter change, save the current total parameter set as the new base point, and go to the next parameter.

(2) If the positive variation of the parameter did not result in a reduction of the objective function, decrease the original value of the parameter by five percent and evaluate the objective function. If the objective function decreases, a new base point is established; if not, reset the parameter to its original value and go on to the next parameter.

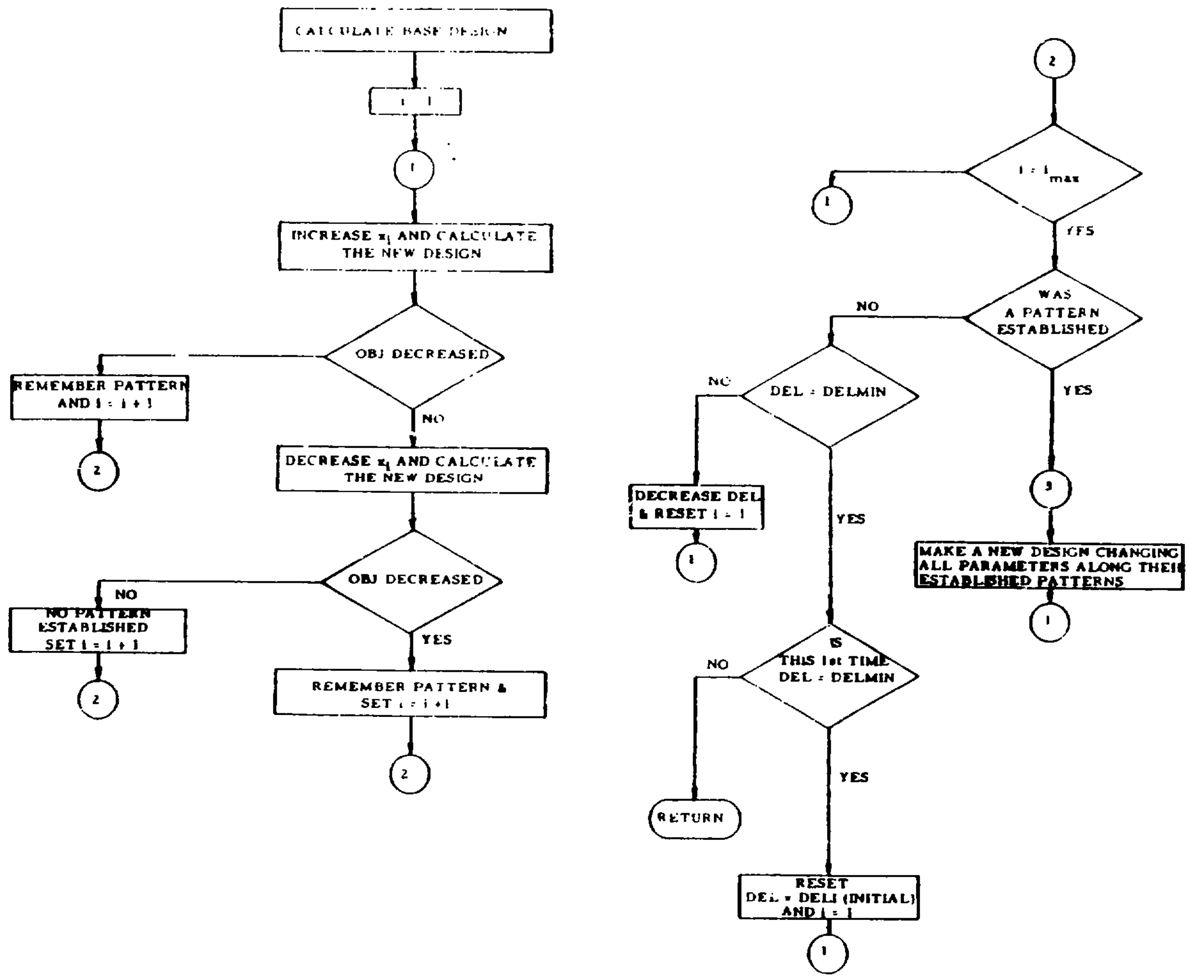


Figure 7. Subroutine PATSH Flow Chart

If the preceding exploratory move for this parameter did not produce a reduction in objective function when the parameter was varied positively, but did when it was varied negatively, then the next exploratory move tries the negative direction first (and then the positive if no improvement is seen).

(3) When all parameters have been varied one at a time, either a new base point will have been established, or the original base point will be retained if none of the exploratory moves resulted in an improvement. If an improvement has been achieved, the exploratory moves have established a pattern - change the first parameter positive, do not change the second parameter, change the third parameter negative, etc. - from which a pattern move can be taken. A pattern step is one in which all parameters producing an improvement during the exploratory moves are varied simultaneously. If no improvement was obtained during the exploratory moves (i.e., the previous base point has been retained), the step size is reduced to one-half its current value and the exploratory moves are repeated.

The pattern step may or may not produce a decrease in the objective function over the current base point. PATSH does not immediately reject a pattern move that results in an increase in the objective function. Each pattern move is followed by another set of exploratory moves, using the pattern move parameter set as the "base" point. If none of these exploratory moves provides a lower objective function value than the base point. value prior to the pattern move, the previous point is retained, and a set of exploratory moves is made about it. If this set does not produce a reduction in OBJ, the step size is reduced for a new set of exploratory moves about the current base point. An improvement in the objective function by any means (exploratory move or pattern move) is always retained as the new base point. The search is assumed to be converged when, through repeated efforts to obtain improvements, the step size is reduced from its original value to the minimum value specified by user input (DELMIN). Such a process may appear to be succeeding by failing to achieve any better point; however, the final set of exploratory moves clearly demonstrates no improvement in the objective function by perturbing all of the parameters in either direction. This is similar to evaluating, through a finite difference method, the first-order partial derivatives of the objective function with respect to the design parameters. Any error in obtaining an optimum would be contained within the minimum step size used for the final exploratory moves.

When using numerical optimization techniques, there is always concern over whether the true, or global, optimum has been reached, or whether a local optimum is the result. No guarantee exists that the solution to a non-linear problem is not a local optimum. The only way to gain a feeling of confidence in the solution (if it is in doubt) is to use different starting points (i.e., different initial (user-supplied) parameter sets), and to determine whether or not the same solution is reached each time. The possibility of local optima is a function of the problem to be solved. Some problems with highly complex constraints may have a number of local optima while many problems have only one global optimum. Keep in mind that, even though the solution may be suspected to be a local optimum, if all penalties are zero, then the solution is a valid design; some improvement in the payoff parameter may be realized, and that can be determined only through starting the search with a different input set.

Performance Requirement, Design Constraint, and Operating Limit Satisfaction

The optimization routine, PATSH, operates by minimizing a single-valued objective function. This single value must reflect the pay-off quantity (which is multiplied by -1 if maximization is desired) and the effectiveness of the design in meeting the performance requirements, design constraints, and operating limits. This has been accomplished by incorporating a penalty function scheme such that

$$OBJ = PAYOFF + \sum_{i=1}^{n} F_i$$

where OBJ is the single-valued objective function minimized by PATSH, PAYOFF is the payoff quantity, and the F_i are individual penalty functions for each of the performance requirements, design constraints, and operating limits (all of which are considered as constraints on the optimization process, and will be referred to as such for the remainder of this discussion). Two basic types of constraints exist, inequality constraints and equality constraints.

Penalty function values (F_i) for violation of a given constraint have the form

$$F_i = g_i^2 \, S_i$$

where g_i = difference between the current value and the constraint value of the i^{th} parameter

S_i = scale factor used to normalize constraint violation penalties to an appropriate level with respect to the payoff parameter.

The choice of this form for the penalty functions provides a penalty value that can be scaled to relatively small values for minor violations with rapidly increasing (second order) value for larger violations. Constraint enforcement in this manner can be thought of as a "soft" constraint (i.e., minor violations are not totally excluded from the solution). Certain limits on design parameter values are enforced as "hard" constraints. An attempt by the optimizer routine to specify a design parameter value which violates a "hard" constraint results in the specified value being overridden with the limiting value and the generation of a penalty function proportional to the attempted violation. An example of a "soft" constraint is the upper limit on propellant web fraction, because a web fraction slightly greater than the limit may be acceptable if it produces greater improvements elsewhere. An example of a "hard" constraint is the length of one part of the motor, because a length of less-than-zero is physically meaningless (and can be computationally misleading).

Adjustable Variables

There are 36 variables in SPOC which may be adjusted by PATSH to obtain an optimum design (Table 1). However, not all of the decision variables can be adjusted in any one given problem because some are peculiar to certain grain geometries. The decision variables fall into these categories

- Propellant grain cross-section dimensions
- Propellant grain lengths
- Propellant ingredient relative weights
- Propellant ballistic characteristics (burn rate and performance level, the latter as influenced by ingredient amounts)
- Nozzle dimensions
- Miscellaneous (motor diameter, case cylindrical wall thickness)

PERFORMANCE REQUIREMENTS

The following performance parameters are driven toward user-input requirements. Penalties are calculated for not meeting each requirement. Default values provided in the code prevent the penalties from being activated unless the user chooses to enforce the requirement. The accompanying parenthetical expressions give the appropriate limit.

- Total impulse (lower three-sigma value at low temperature)
- Total motor weight (maximum nominal)
- Ignition thrust (lower three-sigma value at low temperature)
- Ignition thrust (upper three-sigma value at high temperature)

- o Burn time (lower three-sigma value at high temperature)
- o Burn time (upper three-sigma value at low temperature)
- o Axial acceleration (maximum nominal at high temperature)
- o Change in velocity (minimum nominal value at low temperature)
- o Time-to-target (maximum nominal value at low temperature)
- o Impact (or termination) velocity (minimum nominal value at low temperature)

Those requirements that are shown above to apply to a particular grain temperature condition can also be enforced with a one-temperature problem.

DESIGN CONSTRAINTS AND OPERATING LIMITS

Design constraints and operating limits that are enforced in the SPOC are:

- o Case, closure and nozzle support thickness (sufficient for maximum expected operating pressure plus safety factor)
- o Case and nozzle structure wall thickness ($\geq$ manufacturing limit)
- o Nozzle ablative structural margin of safety (≥ 0)
- o Nozzle ablative thickness ($\geq$ that required for char, ablation and thermal protection)
- o Propellant strain margin of safety during low temperature storage in both CP and valley sections of grain (≥ 0)
- o Propellant strain at low temperature ignition pressurization ($\leq$ input maximum)
- o Propellant web fraction ($\leq$ maximum based on design experience)
- o Propellant thickness under propellant valley ($\geq$ manufacturing limit)
- o Propellant total solids (between maximum and minimum limits)
- o Propellant burn rate and pressure exponent (between maximum and minimum limits)

- Burn rate catalyst and fuel contents (≤ maximum based on experience)
- Combustion gas Mach number in port at low temperature (nominal ≤ maximum based on experience)
- Chamber pressure at high temperature (nominal ≤ maximum based on experience)

- Geometrically valid (compatible) propellant grain cross-section dimensions
- Lengths and thicknesses greater than zero
- Motor dimensions (length ≤ maximum, nozzle exit diameter ≤ maximum, case aft opening radius = nozzle entrance radius, nozzle blast tube length and diameter = requirement)
- Geometrically valid (compatible) nozzle dimensions
- Longitudinal combustion stability

PAYOFF PARAMETERS

The PAYOFF parameters from which the user can select one to be minimized during any given machine submission are

- None
- Total motor cost (minimize)
- Total motor weight (minimize)
- Total impulse (maximize)[1]
- Total impulse-to-total motor weight ratio (maximize)[1]
- Burnout Velocity (maximize)[1]

(1) PATSH will minimize the product of minus one times the value of this parameter, which produces a maximization of the parameters.

LIMITATIONS AND ACCURACY

The purpose of this discussion is to summarize some general limitations of the code and to provide estimates of the accuracy of the results.

Limitations

Some limitations on the use of SPOC are inherent in the assumptions employed during original development of the analysis and simulation modules; these assumptions are given in the discussions of the individual modules (Reference 4) and their impact on a given problem solution is best left to the user.

Basically, there are no restrictions on the size of the motor which may be analyzed with SPOC. Small motors operating at high pressure could possibly enter the regime where thin-wall pressure vessel equations should be replaced by thick-wall relationships; it is up to the user to recognize this situation. The cylindrical section of a motor employing elliptical closures (forward or aft, or both) cannot be reduced to zero length because of how the grain geometry is described to the ballistic simulation module; the minimum length attainable is between one and two grain web thicknesses. As for large motors, there are no restrictions.

All volumes and concomitant weights are calculated from exact geometric relationships; there are no internal empiricisms to estimate weights. Weights not amenable to direct calculation in an optimization code (e.g., igniter, safe-and-arm device, wings, etc.) are user-supplied values.

Of necessity, some of the analysis routines are somewhat simplified, as would be expected when operating in a preliminary design mode; however, all analysis routines are industry-accepted methods.

(a) Propellant strain is calculated under plane-strain conditions. Thus end-effects and three-dimensional effects during rapid configuration changes are not accounted for.

(b) Membrane stresses in the ellipsoidal pressure vessel closures (Type 1) are calculated at the motor centerline which provides a satisfactory estimate of the required closure thickness elsewhere. Bending stresses at the closure-to-cylindrical shell junction are not considered.

(c) Bending at the closure-to-cylindrical shell junction is considered for the Type 3 forward closure (that features a flat plate closure integral with the cylindrical shell) as long as material response is elastic. Transitions between the cylindrical shell and the integral flat plate (i.e., radii or gradually increasing cylindrical wall thickness in the vicinity of the closure) are not included in stress estimates or volume calculations.

(d) The user must input a heat-transfer coefficient for each of the three nozzle ablative materials, which means that the coefficient is constant for all flow conditions to which a particular material is exposed.

There are dimensional mismatches at case-to-closure tangent points and case-to-nozzle joints in order to allow the user complete flexibility in choosing his motor arrangement and to make the computations more manageable; however, the results of these mismatches on predicted ballistic performance and weights is thought to be minimal. Figure 8 shows the potential mismatch between the liner inner surface at the closure-to-case cylindrical section interface; there are two ways that this mismatch can occur, and both are considered when the grain outer dimensions are established for the ballistic simulation. Figure 9 shows the potential mismatch of the pressure vessel outer surface at the closure-to-case cylindrical section interface. The inner surfaces of the closure and cylindrical section exactly match at the tangent point. Then the required closure thickness (TCLOF) is calculated after the ballistic simulation, and the cylindrical section thickness (TCASE) is a PATSH-adjusted parameter that eventually is satisfactory for the maximum pressure. Thus the outer surface of the pressure vessel could have a discontinuity at the tangent point. The thrust skirt is also shown in Figure 9 to show that its mating surface is the cylindrical section outer surface. Obviously, the degrees of mismatch shown in Figures 8 and 9 are greatly exaggerated for clarity; their effects on weights is negligible.

Another mismatch that always occurs is shown in Figure 10. The case opening radius (RNOZEN) always (eventually) is equal to the nozzle entrance radius, so there is no mismatch there. However, the nozzle ablative and structural support calculations are performed normal to the internal surface, so that part of the nozzle coincides with the case as shown by the shaded area in Figure 10; this "duplication" of volume provides an allowance for the nozzle attachment flange.

The trajectory simulation employs a point-mass missile flying a two-dimensional path in the altitude-range plane over a flat earth. Forces modeled are restricted to thrust, drag and weight (i.e., lift is always zero), and angle of attack is always zero. The trajectory simulation is intended as a supplementary evaluation tool (unless, of course, this model accurately describes the problem under consideration).

A two-dimensional plane-strain model is used to calculate propellant strain due to low-temperature storage and ignition pressurization. Such a model accurately describes the propellant behavior at a point mid-way along the grain length when the grain length-to-diameter ratio (L/D) is equal to or greater than about seven. For $L/D < 7$, or for locations near the grain terminations, the plane-strain models give very conservative predictions because the end effects (three-dimensional) that relieve the strain are not accounted for in SPOC. Strains predicted for a propellant valley or slot will also be conservative near the ends or for short slots.

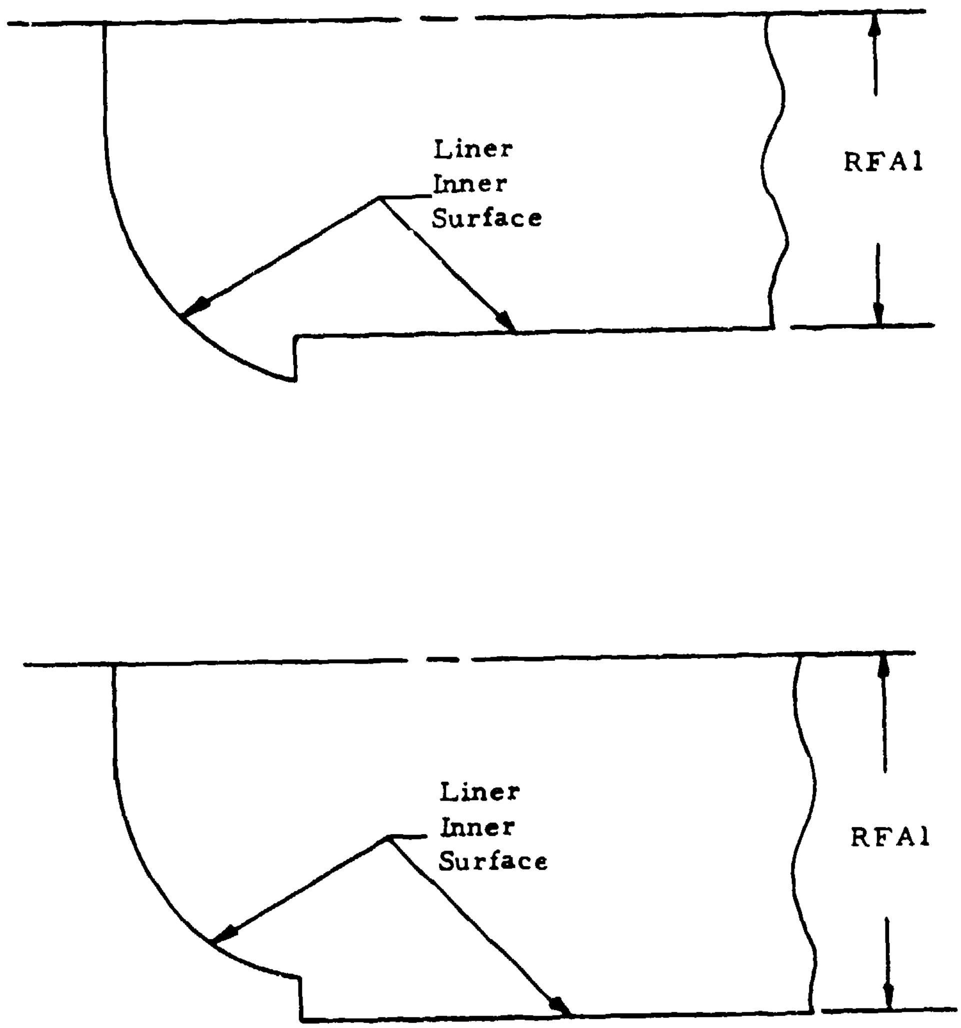


Figure 8. Potential Dimensional Mis-match of Liner Inner Surface

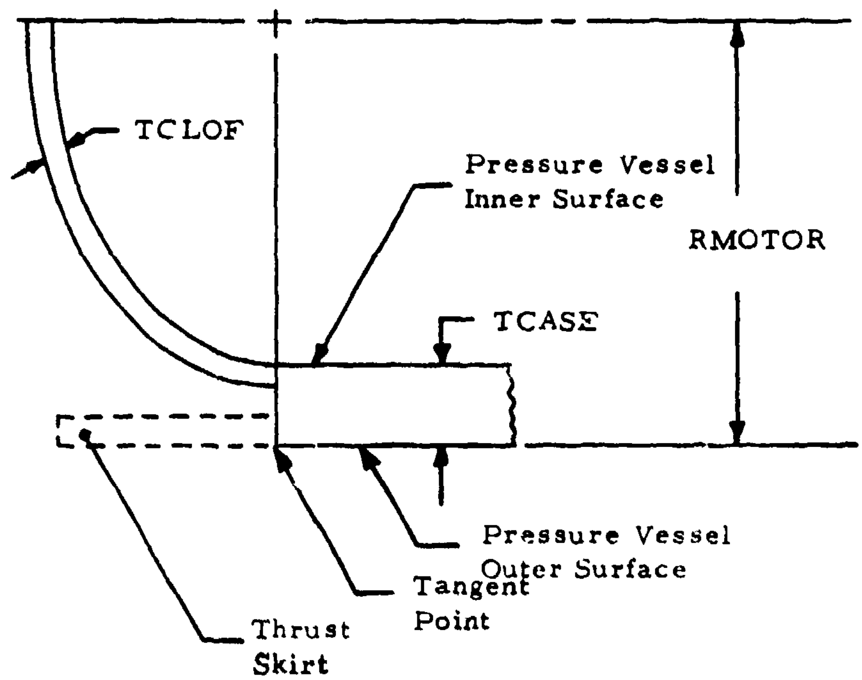


Figure 9. Potential Dimensional Mis-match of Pressure Vessel Outer Surface

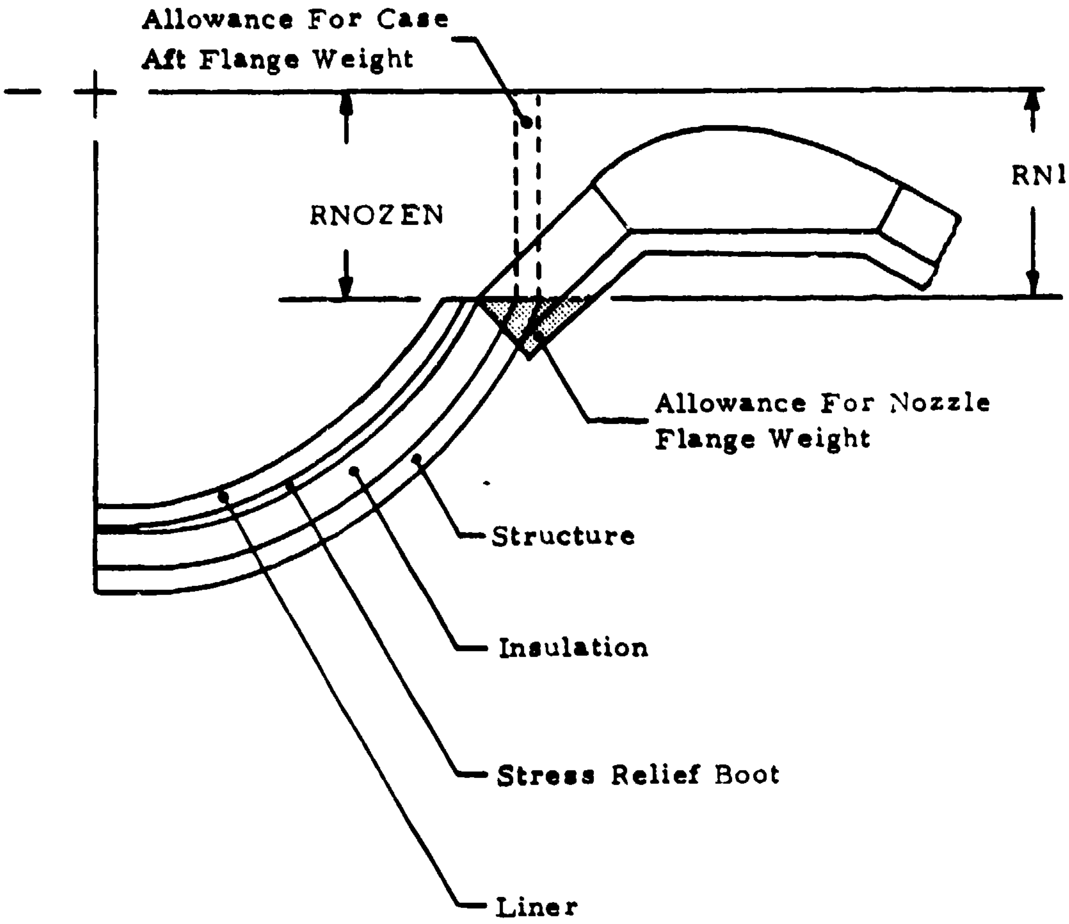


Figure 10. Dimensional Mismatch at Case-to-Nozzle Interface

The propellant structural analysis is not conservative at the hinge points of stress relief flaps and at the transition between propellant lots and CP regions. Both of these areas represent highly three-dimensional conditions that are not amenable to preliminary design calculations used in SPOC. Consequently, there is the inherent assumption that the bore conditions are the critical locations. Provisions have been made to include volume and weight allowances for stress relief boots in ellipsoidal closures, even though their final configuration is dependent on more detailed analyses.

Thermal strain in the propellant due to low-temperature storage is compared with design strain endurance (nominal strain endurance reduced for mix-to-mix variations and aging degradation). Strain induced by ignition pressurization is compared with a user-input maximum limit. This latter limit should be derived from tests that measure strain capability at rapid strain rate (to simulate ignition pressurization) on test specimens conditioned to the design low temperature and already strained to the level that will be induced by low temperature storage.

Accuracy of Code

There are three levels of accuracy to consider in the evaluation of a computer code. First, the user must decide how well the mathematical equations model the reality of a particular problem. Second is the computational accuracy, or how faithfully the programmer has carried out the mathematical manipulations. Finally, and totally under the control of the user, is the accuracy of the input data. Only the first two levels will be discussed here.

Accuracy of the mathematical models is paramount in the overall accuracy of a code. The several analysis and simulation modules are discussed separately in the following list:

Module	Estimated Accuracy of Model
Ballistic simulation	± 3% total impulse ± 5% maximum pressure General qualitative assessment based on experience
Weight estimates	± 2% General qualitative assessment based on experience
Propellant theoretical characteristics	Essentially error free. Uses NASA-Lewis thermochemical analysis (Reference 13).
Combustion stability	Based on AFRPL Standard Stability Code (Reference 7).

Module	Estimated Accuracy of Model
Combustion efficiency	Based on AFRPL Solid Propellant Prediction Code (Reference 5).
Motor costs	Based on Tri-Services Rocket Motor Trade-off Study for steel cases (Reference 6).
Trajectory simulation	Estimated to be very high, provided the problem is adequately described by the model. See discussion above.
Propellant structural analysis	Strain calculation "very accurate" in center of motor with $L/D > 7$ (probably within 10%). For location near ends of long motor or for $L/D < 7$, calculated strains are conservative, with degree of conservatism depending on problem.
Pressure vessel structural analysis	Estimated to be conservative by approximately 15%.

The computational accuracy of the code is extremely high. Iteration schemes in the ballistics simulation and grain subroutines require convergence to within 0.01% or less. The trajectory simulation uses an industry-accepted technique. Thus it is felt that the mathematical models have been faithfully computed.

SUMMARY AND CONCLUSIONS

The project objective was met. A computer code (SPOC) was developed to perform detailed preliminary designs of a wide variety of solid rocket motors and to provide for the optimization of those designs. The code is operational on IBM and CDC computers.

SPOC has the capability of analyzing five different propellant grain geometries, three forward and two aft closure arrangements, and six nozzle configurations in any combination. The ballistic simulation module includes mass addition and erosive burning effects and an ablating throat model; the burning surface history is internally generated with a rigorous geometric regression from source dimensions. Weights and lengths are calculated from source dimensions except for components such as igniter, safe-and-arm device, wing clips, tunnels, environmental closures, etc; the latter are user-inputs. Propellant characteristics are calculated with the NASA-Lewis thermochemistry code or can be furnished by the user. Longitudinal combustion stability margins can be analyzed and included in the motor optimization considerations. A simplified trajectory simulation can be performed. Other design calculations provided are pressure vessel structural analysis, propellant structural analysis, motor cost, and nozzle ablation, thermal and structural analyses.

A three-volume User's Manual was prepared and published as a separate document (Reference 4). The Manual includes complete instructions for operating the code and a detailed description of the analysis routines.

A briefing was held for members of the propulsion community. Representatives of prime and propulsion contractors were provided the User's Manual, a sample case input/output listing, briefing notes, and a magnetic tape copy of the code.

Thiokol and AFRPL personnel held face-to-face review meetings on a regularly scheduled basis to insure the code provided the features and functions desired by AFRPL.

Five sample design problems furnished by AFRPL were satisfactorily solved with SPOC using the AFRPL computer. These problems used various options of the code, and their solution demonstrated the adequacy of the analyses.

SPOC was designed and implemented with ease of operation for the user being one of the primary objectives. Performance requirements, design constraints and operating limits are not invoked in the optimization search unless user-specified. All inputs are defaulted to "safe" values so the user must provide only those inputs necessary for a given problem; certain inputs are identified in the Manual as being "required", meaning that the problem cannot be completely defined without them. All inputs (except

for only two exceptions) are in namelist format and they are divided into logical groups by subject. The complete namelist series is printed (including default values) as part of the output. Certain analyses are not performed unless specified by the user (e.g., combustion stability, trajectory simulation, propellant structural analysis, motor cost analysis, thermochemical calculations, impulse efficiency calculation). There is internal checking for the compatibility of constraints and limits and for the problem definition. Messages are printed whenever there is an abnormal termination to describe the situation and to suggest changes. The current set of decision variables is also printed in the event of an abnormal termination. The code internally selects (a) yield or ultimate strength design conditions, whichever is more critical; (b) minimum length contour for the nozzle expansion section; (c) critical stress condition in nozzle support structure; (d) coordinates to define propellant geometry in pressure vessel end closures; (e) coordinates of internal nozzle surface.

Additional versatility is provided in SPOC through several features. User-supplied models can be employed for propellant burn rate, propellant strain endurance, propellant rheological property, motor cost, impulse efficiency, and propellant combustion response. Thermochemical properties of up to four new ingredients can be added by the user to the menu built into the code. There is complete interchangeability of motor components. The user can specify desired analyses for a given problem.

The computer code execution time is strongly dependent on the user and the problem. Factors affecting execution time are the number of decision variables in the optimization process, which analyses are specified, time step size in ballistic and trajectory simulations, length step size in nozzle analysis, number of modes in stability analysis number of propellant ingredients, trajectory termination, number of grain temperatures for ballistic simulation, minimum step size and number of base point iterations in optimization process.

RECOMMENDATIONS

The following recommendations are made as a result of the activities on this project.

1. Include in all computer code development projects a task equivalent to the Phase I, Task 1 (Determination of Level of Detail) effort on this project. The task should require a detailed description of the planned code. Show models in as much detail as possible, consistent with the task being completed during the first 10% (approximately) of the project calender time. Document results of this task; the contractor's informal report system would be appropriate. However, PCO approval of the report (plan) should be required prior to project continuation, and all future project activities should be based on this plan. Deviations from the plan should be formally documented between AFRPL and the contractor.

2. Hold face-to-face review meetings between appropriate contractor and AFRPL personnel at regular intervals during model formulation and code preparation tasks of a computer code development project. These should be "working meetings", with detailed information exchange being the primary objective. Purpose of the meetings is to insure that the code details being developed by the contractor are consistent with what AFRPL is expecting.

3. Add the following capabilities to the current version of the code:

a. Additional propellant grain geometries (aft-slotted finocyl; radial (transverse) slots; combinations such as CP/star, CP/wagon wheel, etc.; forked wagon wheel (dendrite)).

b. Two-level thrust ballistic definitions and optimization control.

c. Pulse motor grain analyses.

d. Expanded selection of propellant ingredients.

e. Improved version of PATSH or a different optimization technique to reduce the computer execution time needed to obtain an optimum solution.

f. Improved nozzle ablation analyses.

g. Two-propellant combinations.

h. End effects in propellant structural analysis.

4. Establish a "maintenance" contract to incorporate revisions to code and manual as more experience is gained with the code.

REFERENCES

1. AFRPL-TR-80-65, "Low Cost Missile Motor Demonstration Program, Final Report", G. P. Roys, et al, Thiokol/Huntsville Division, November 1980.

2. U-80-354, "Advanced Air Launched Missile Motor Design Methods: Phase I, Task 1 Report", G. P. Roys and D. D. Maser, Thiokol/Huntsville, March 1980.

3. U-80-4464, "User's Manual for E469 Motor Design Computer Code", G. P. Roys, Thiokol/Huntsville, February 1980.

4. AFRPL-TR-81-70, "User's Manual for Solid Propulsion Optimization Code (SPOC)", Volume I - Technical Description, Volume II - User's Guide, Volume III - Program Description, G. P. Roys, Thiokol/Huntsville, October 1981.

5. AFRPL-TR-75-36, "A Computer Program for the Prediction of Solid Rocket Motor Performance, Volume I", Ultrasystems Inc., Irvine, CA. July 1975.

6. CPIA Publication 322, "Tri-Services Rocket Motor Trade-off Study", Booz, Allen Applied Research, Contract N00123-76-C-0732, May 1980.

7. AFRPL-TR-76-32, "Standardized Stability Prediction Method for Solid Rocket Motors", R. L. Lovine, May 1976.

8. AFRPL-TR-YY-YYY, "Tactical Missile Optimization Program-TACMOP", Final Report, M. D. Harsh Hercules/ABL, October 1979.

9. Whitney, D. E., Personal Communication. Joint Mechanical and Civil Engineering Computing Facility, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139.

10. Hooke, R., and T. A. Jeeves. "Direct Search Solution of Numerical and Statistical Problems". Journal of the Association for Computing Machinery, Vol. 8, No. 2. pp 212-229.

11. Afimiwala, K. A., and Mayne, R. W., "Evaluation of Optimization Techniques for Applications in Engineering Design". Journal of Spacecraft and Rockets, Vol. II, No. 10. pp 673-674.

12. Eason, E. D., and Fenton, R. G., "A Comparison of Numerical Optimization Methods for Engineering Design", Transactions of the ASME, Ser. B: Journal of Engineering for Industry, Vol. 96, February 1974, pp 196 - 200.

13. Svehla, R. A., and McBride, B. J., "FORTRAN IV Computer Program for Calculation of Thermodynamic and Transport Properties of Complex Chemical Systems", NASA TN D-7056, Lewis Research Center, January 1973.